

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра САПР**

**КУРСОВАЯ РАБОТА**  
**по дисциплине «Сети ЭВМ»**  
**Тема: Игра "Шахматы" с передачей данных по UDP**

Студент гр. 1302

\_\_\_\_\_

Новиков Г.В.

Преподаватель

\_\_\_\_\_

Горячев А.В.

Санкт-Петербург

2024

## **ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ (КУРСОВОЙ ПРОЕКТ)**

Студент Новиков Г.В.

Группа 1302

Тема работы: Игра "Шахматы" с передачей данных по UDP

Исходные данные:

Язык Python 3.12.2, библиотеки chess, socket, tkinter

Протокол UDP

Содержание пояснительной записки:

«Аннотация», «Содержание», «Введение», «Теория», «Архитектура приложения», «Пример работы приложения», «Заключение», «Список использованных источников»

Предполагаемый объем пояснительной записки:

Не менее 15 страниц.

Дата выдачи задания: 02.04.2024

Дата сдачи реферата: 03.06.2024

Дата защиты реферата: 03.06.2024

Студент

\_\_\_\_\_

Новиков Г.В.

Преподаватель

\_\_\_\_\_

Горячев А.В.

## **АННОТАЦИЯ**

В данной курсовой работе была реализована программа для игры в шахматы по локальной сети с пользовательским интерфейсом. Для передачи данных был использован протокол UDP, реализовано подключение двух устройств (ПК) друг к другу с помощью библиотеки socket. Пользовательский интерфейс реализован с помощью библиотеки tkinter. Результатом работы является исполняемый файл.

## **SUMMARY**

In this course work, a program for playing chess over a local network with a user interface was implemented. The UDP protocol was used for data transmission, enabling the connection of two devices (PCs) to each other using the socket library. The user interface was implemented using the tkinter library. The result of the work is an executable file.

# СОДЕРЖАНИЕ

## ОГЛАВЛЕНИЕ

<b>ВВЕДЕНИЕ.....</b>	<b>5</b>
<b>1. ТЕОРИЯ.....</b>	<b>6</b>
1.1. Игра "Шахматы" .....	6
1.2. User Datagram Protocol.....	6
<b>2. АРХИТЕКТУРА ПРИЛОЖЕНИЯ.....</b>	<b>7</b>
2.1. UDPClient.....	7
2.2. Game.....	7
2.3. UI .....	8
2.4. App.....	8
<b>3. ПРИМЕР РАБОТЫ ПРИЛОЖЕНИЯ.....</b>	<b>9</b>
3.1. Главная страница .....	9
3.2. Создание игры.....	9
3.3. Подключение к игре .....	12
3.4. Игра .....	14
<b>ЗАКЛЮЧЕНИЕ.....</b>	<b>19</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....</b>	<b>20</b>
<b>ПРИЛОЖЕНИЕ А. ССЫЛКА НА РЕПОЗИТОРИЙ GITHUB.....</b>	<b>21</b>

## **ВВЕДЕНИЕ**

Целью работы являлось создание программы для соединения двух устройств по сети с помощью UDP и дальнейшей игры в шахматы между пользователями этих устройств. Для комфортной игры был реализован пользовательский интерфейс, состоящий из нескольких окон: для создания игры, подключения к игре и самой игры, а также для ожидания подключения соперника.

## **1. ТЕОРИЯ**

### **1.1. Игра "Шахматы"**

В обычные правила игры были внесены некоторые изменения для упрощения реализации

1. Пешка может превратиться только в ферзя
2. Не используется правило 50 ходов, то есть игроки не могут потребовать ничью, если на протяжении последних 50 ходов каждого игрока ни одна фигура не была взята и ни одна пешка не сделала хода (при 75 таких ходах ничья объявляется автоматически по обычным правилам)
3. Не используется правило, при котором в случае троекратного повторения позиции игрок может потребовать ничью (при 5 повторениях ничья объявляется автоматически по обычным правилам)

Для обмена данными между устройствами используется формат записи ходов Universal Chess Interface (uci)

### **1.2. User Datagram Protocol**

Протокол UDP определен для обеспечения режима передачи датаграмм в пакетно-коммутируемой среде компьютерных сетей. Этот протокол предполагает использование протокола IP в качестве базового протокола.

Данный протокол предоставляет процедуру для отправки сообщений приложениями другим программам с минимальным использованием механизмов протокола. Протокол ориентирован на транзакции, и гарантии доставки и защиты от дублирования не предоставляются.

## 2. АРХИТЕКТУРА ПРИЛОЖЕНИЯ

Программа состоит из 4 главных классов:

1. UDPClient – класс для взаимодействия по сети
2. Game – класс для игровой логики, содержит объект UDPClient
3. UI – класс для взаимодействия пользователя с программой
4. App – класс, объединяющий пользовательский интерфейс (UI) и игру (Game)

### 2.1. UDPClient

Позволяет создать соединение между двумя хостами и обеспечивает отправку данных между ними. Использует библиотеку socket для привязки IP-адреса и порта, отправки и получения данных. Многопоточность обеспечена с помощью библиотеки threading. Также использована библиотека netifaces для получения адресов, привязанных к разным сетевым интерфейсам.

Подключение осуществляется следующим образом:

1. Хост 1 привязывает IP и порт и ожидает команды подключения
2. Хост 2 привязывает IP и порт и отправляет команду подключения по адресу и порту, указанным пользователем (должны совпасть с адресом и портом хоста 1)
3. Хост 1 получает команду подключения, сохраняет адрес и порт хоста 2 и отправляет команду принятия хосту 2
4. Хост 2 получает команду принятия, сохраняет адрес и порт хоста 1

### 2.2. Game

Класс, определяющий логику игры. Проверка правильности ходов, а также варианта окончания игры осуществляется с помощью сторонней библиотеки chess. Использует класс UDPClient для взаимодействия по сети. При начале игры игроком, создавшим игру, отправляет команду начала игры противнику. На каждом ходу отправляет ход противнику в формате uci.

### **2.3. UI**

Класс, определяющий пользовательский интерфейс приложения. Задает внешний вид интерфейса, также задает привязку кнопок к командам. Для отображения фигур использованы символы фигур в Unicode

### **2.4. App**

Главный класс приложения, через который приложение запускается. Связывает классы Game и UI.



### 3. ПРИМЕР РАБОТЫ ПРИЛОЖЕНИЯ

В качестве примера рассмотрим подключение и игру с двух разных устройств (ноутбук и ПК), находящихся в одной сети.

#### 3.1. Главная страница

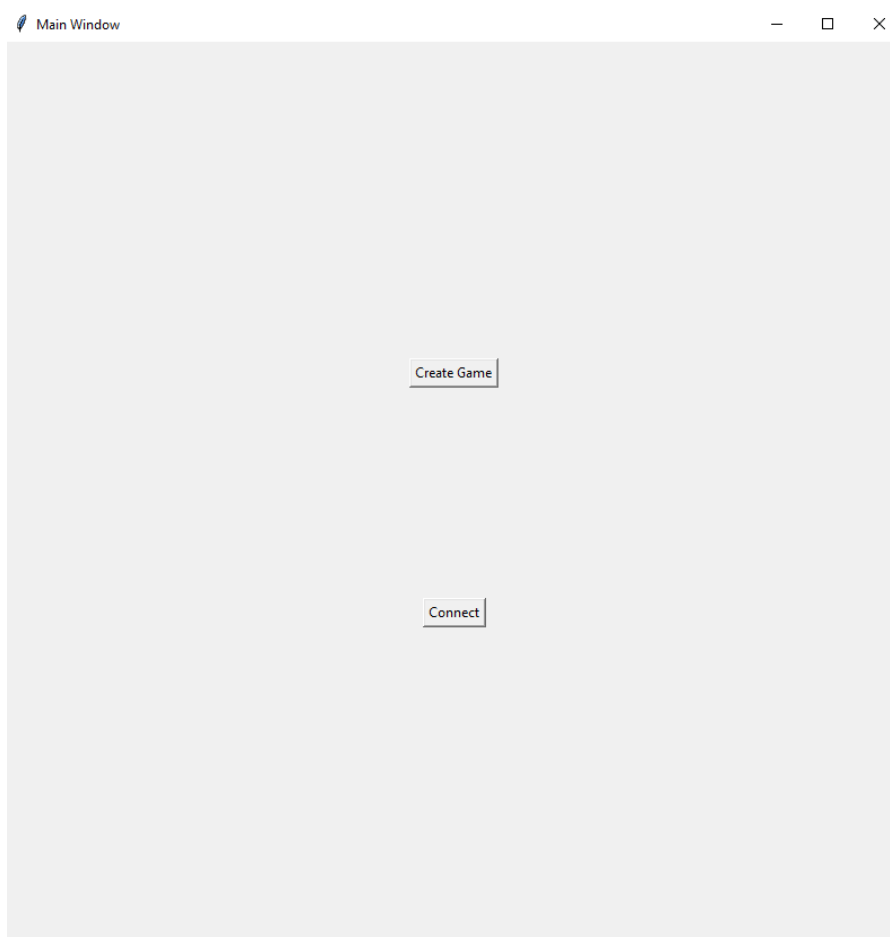


Рис. 1. Главная страница

#### 3.2. Создание игры

При создании игры пользователь может выбрать IP-адрес из списка адресов, привязанных к разным сетевым адаптерам

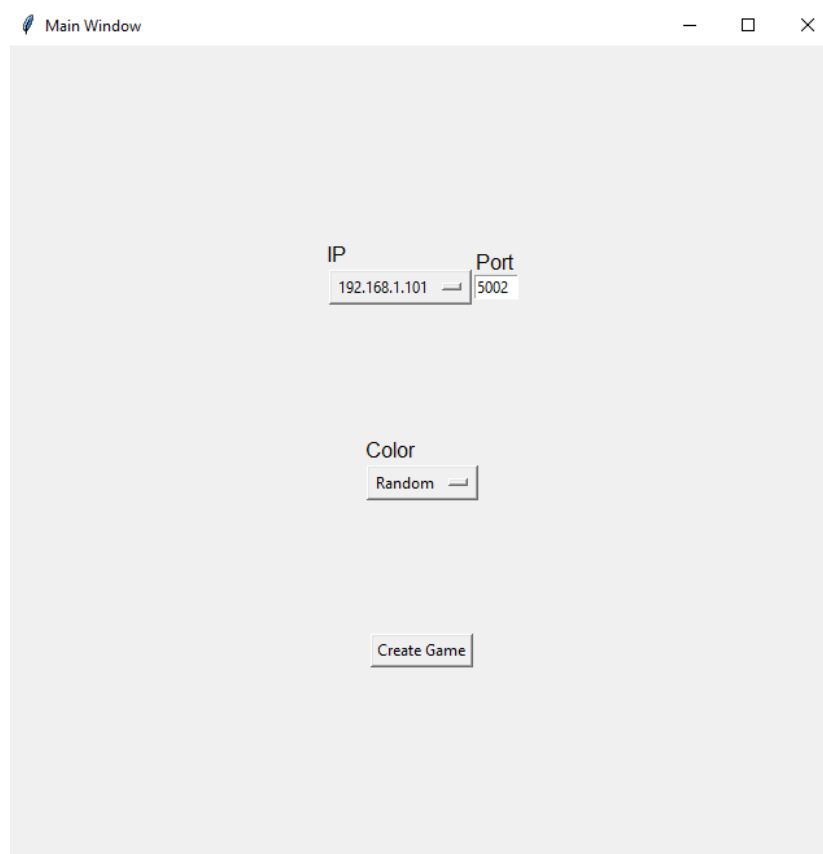


Рис. 2. Создание игры

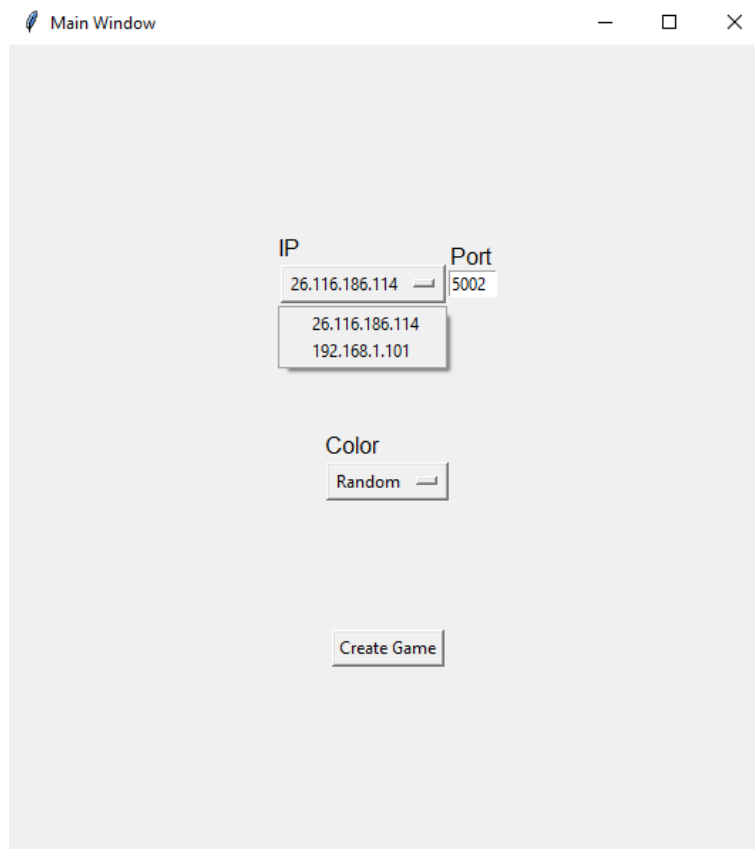


Рис. 3. Создание игры (выбор адреса)

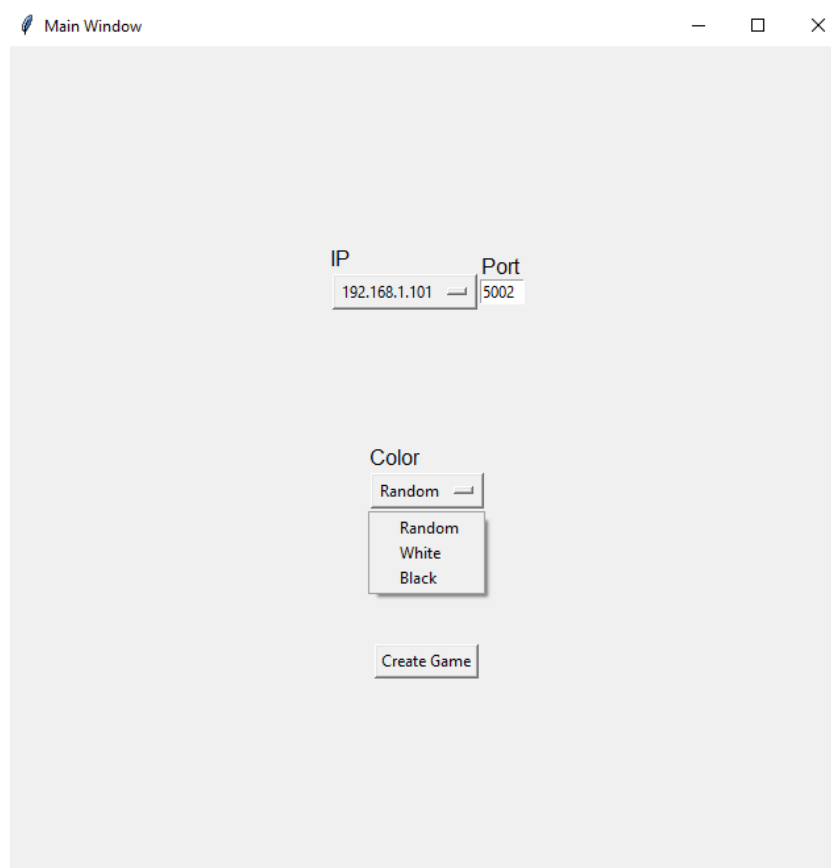


Рис. 4. Создание игры (выбор цвета)

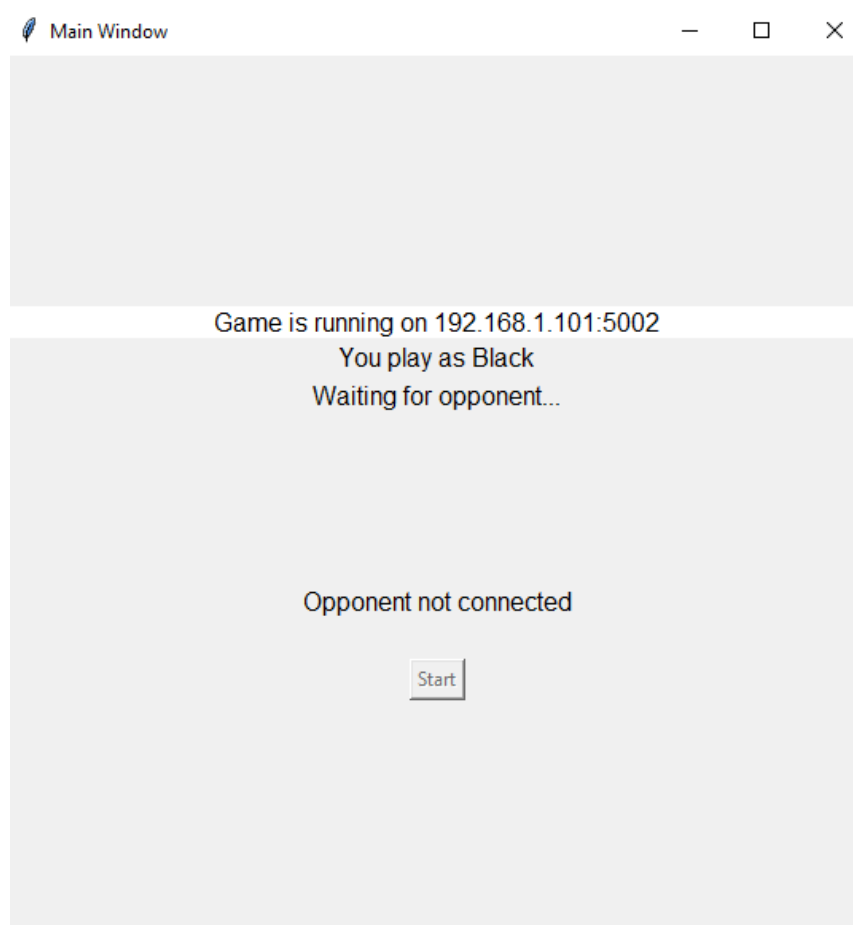


Рис. 5. Ожидание противника после создания игры

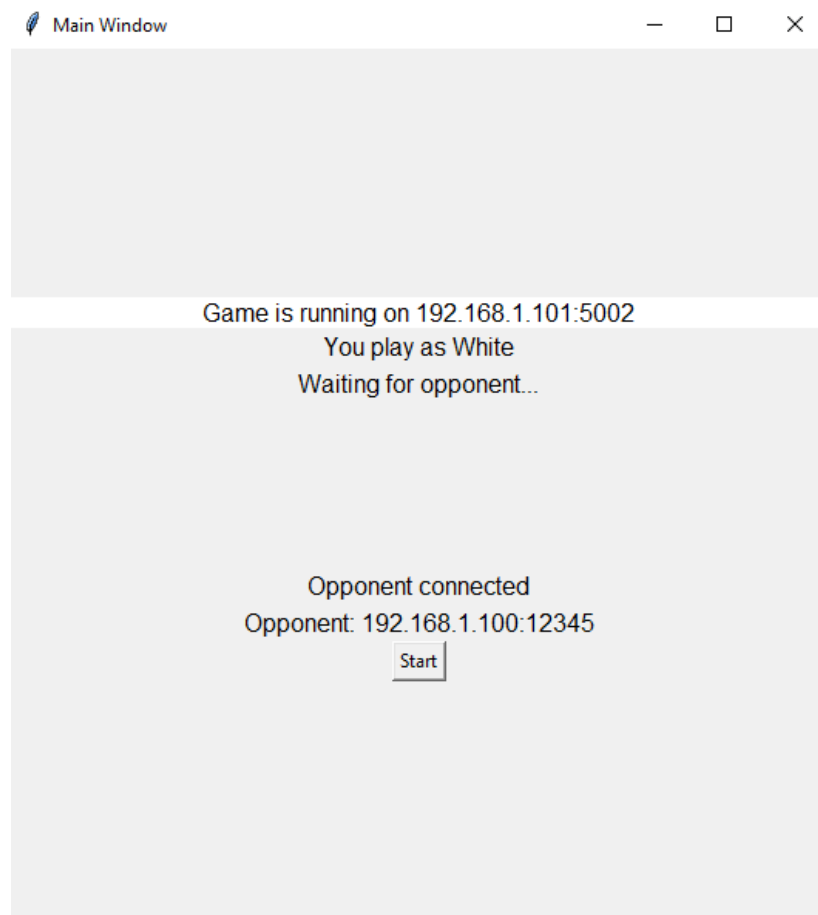


Рис. 6. Окно ожидания после подключения противника

### 3.3. Подключение к игре

При подключении также можно выбрать свой IP-адрес и порт

Main Window

Your ip and port

IP	Port
192.168.1.100	5002

Opponent's ip and port

IP	Port

Connect

Рис. 7. Подключение

Main Window

Your ip and port

IP	Port
192.168.1.100	12345

Opponent's ip and port

IP	Port
192.168.1.101	5002

Connect

Рис. 8. Подключение

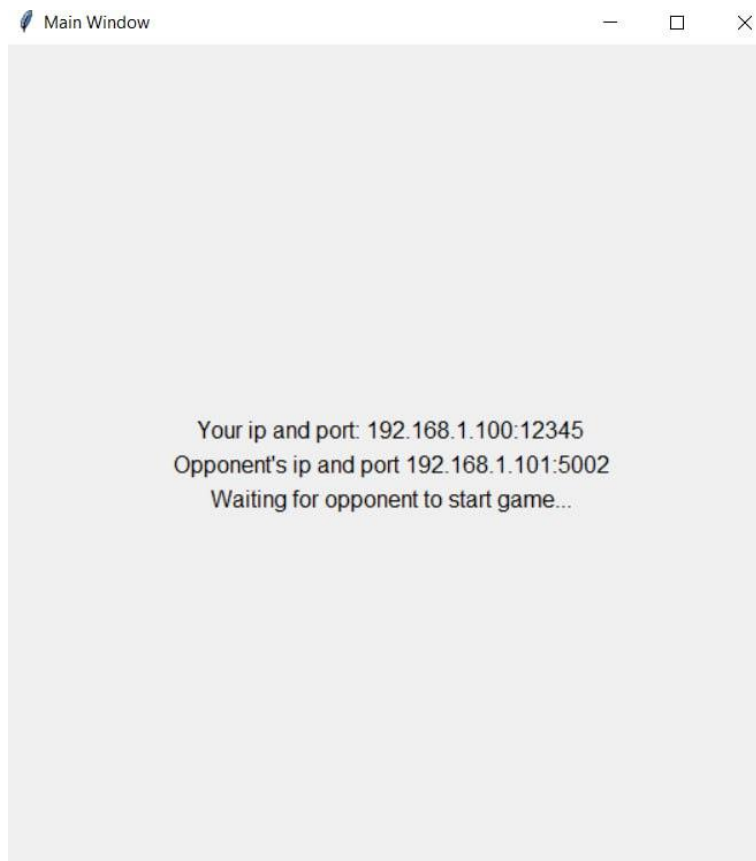


Рис. 9. Ожидание начала игры после подключения

### 3.4. Игра

После начала игры пользовательский интерфейс на обоих устройствах одинаковый

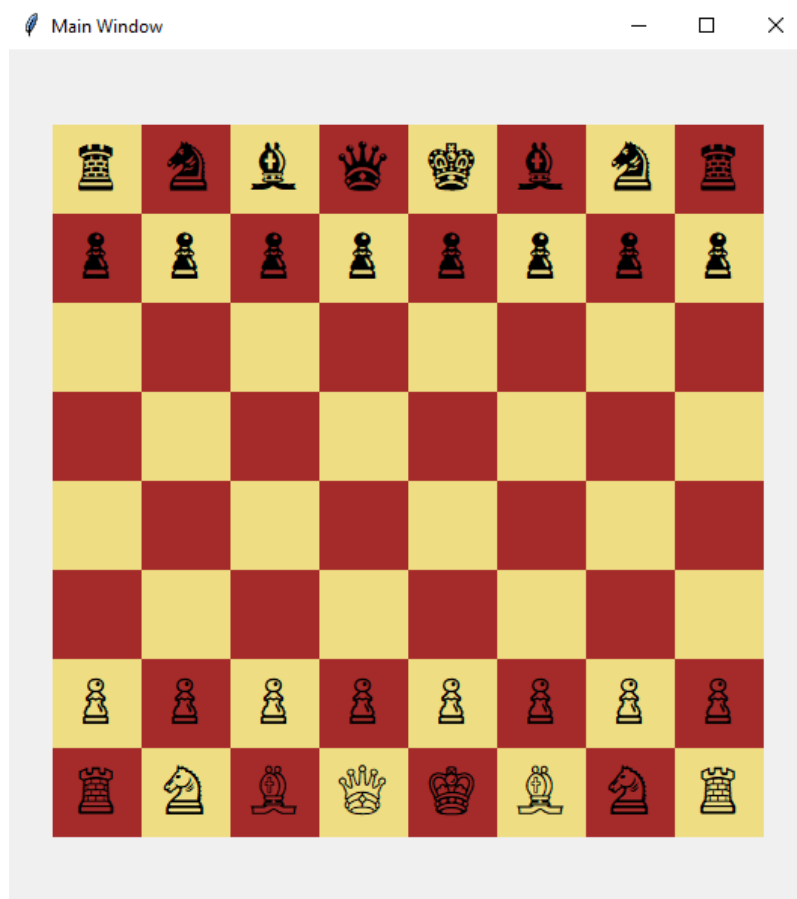


Рис. 10. Начальная позиция

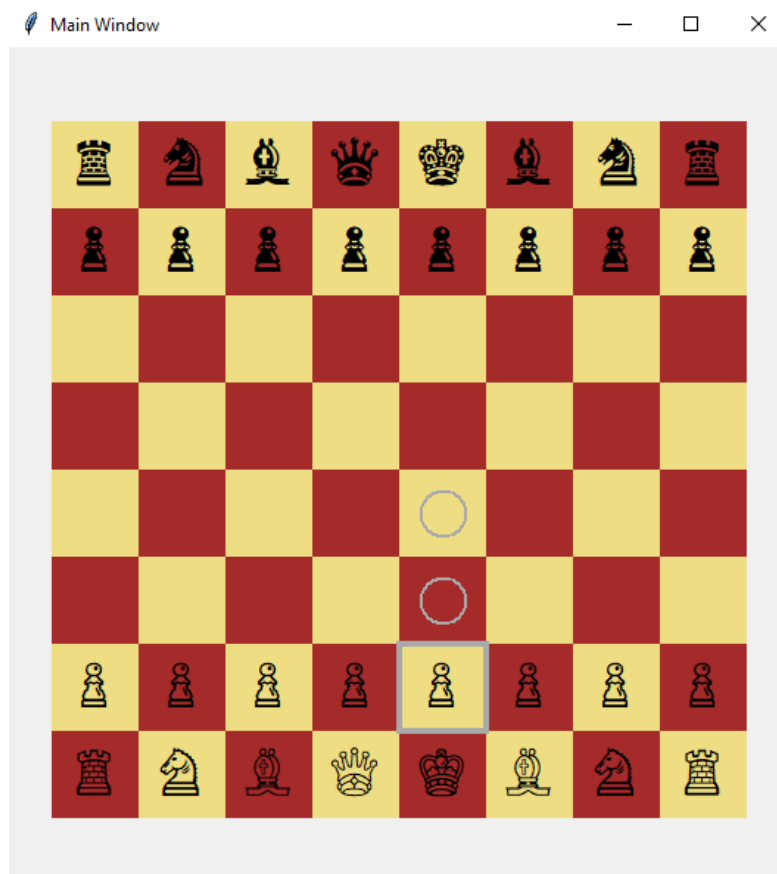


Рис. 11. При выборе клетки

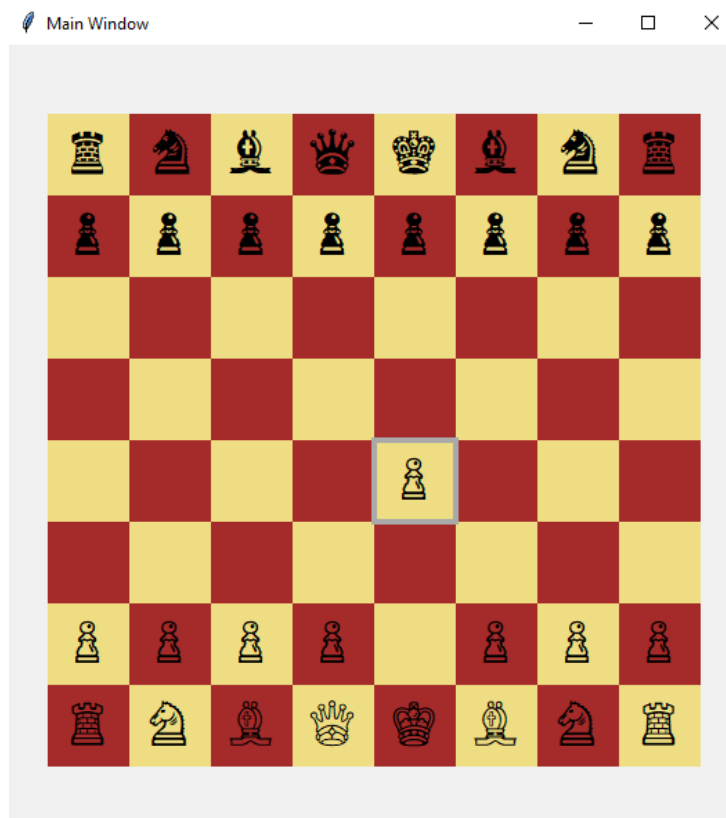


Рис. 12. Ход

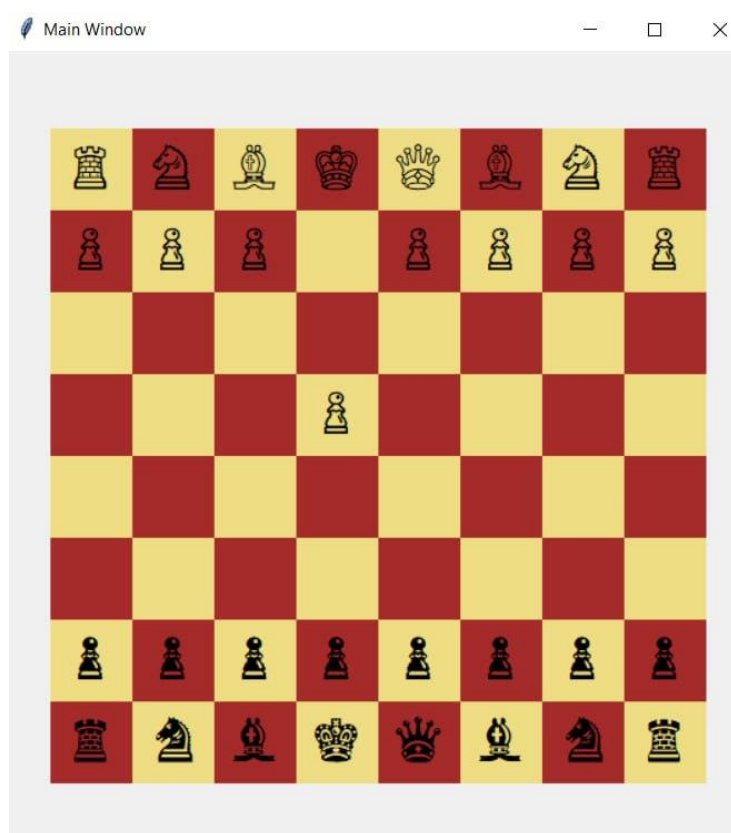


Рис. 13. На другом устройстве



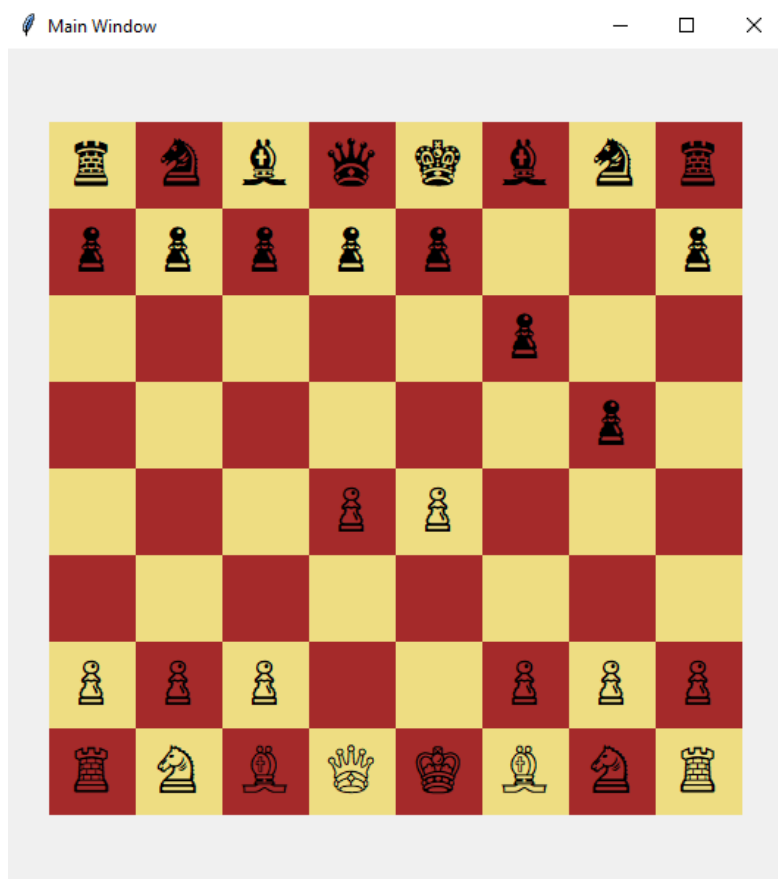


Рис. 14. Доска после нескольких ходов

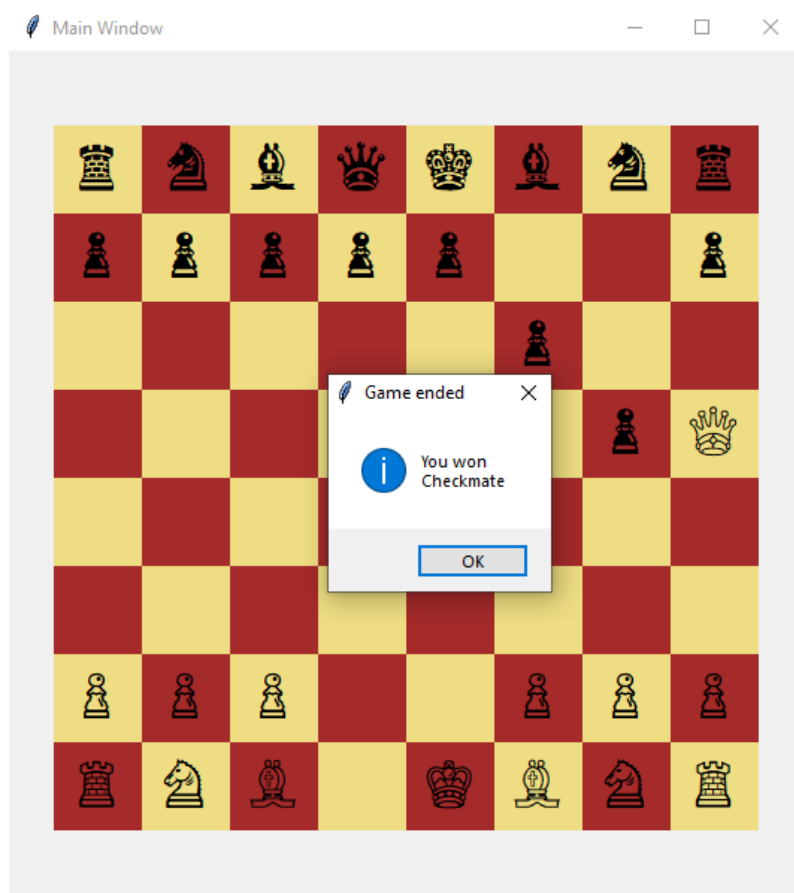


Рис. 15. Сообщение о победе

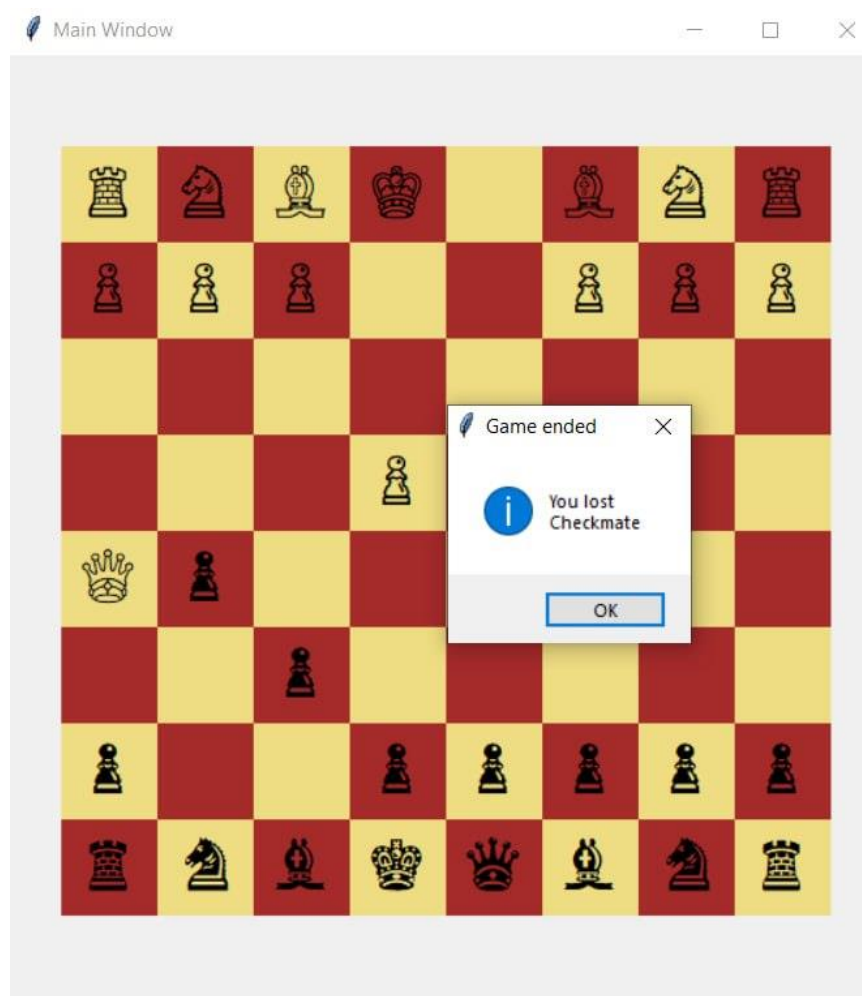


Рис. 16. Сообщение о поражении

Поддерживаются также сообщения о ничьей с указанием причины.

## ЗАКЛЮЧЕНИЕ

В результате данной работы было создано приложение для игры в шахматы по локальной сети с пользовательским интерфейсом. В функции приложения входит создание соединения между двумя хостами, передача данных (ходов и команд для подключения и начала игры) по UDP, создание и настройка игры и сама игра со вводом данных с помощью пользовательского интерфейса.

Программа устанавливает соединение между двумя хостами, однако главным ее недостатком является отсутствие гарантии получения пакетов (ходов) противником. UDP не предоставляет такой гарантии, поэтому она должна быть реализована на уровне приложения. Это можно сделать, например, следующим образом: при отправке пакета приложение ждет ответа от получателя и по истечении какого-то времени, если ответ не был получен, отправляет пакет еще раз. Соответственно, получатель при получении пакета должен отправить подтверждение получения отправителю. В таком случае есть шанс, что один пакет будет получен несколько раз, но в данном случае это не страшно, потому что ход будет применен при первом получении и во второй раз не будет действителен.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Правила игра в шахматы ФИДЕ от 1 января 2023 года // URL: <https://handbook.fide.com/files/handbook/LawsOfChess2023Russian.pdf> (дата обращения: 1.06.2024)
2. Спецификация RFC 768. User Datagram Protocol. Jon Postel, 1980 // URL: <https://datatracker.ietf.org/doc/html/rfc768>
3. Документация языка Python // URL: <https://docs.python.org/3.12/>
4. Документация библиотеки tkinter // URL: <https://docs.python.org/3/library/tkinter.html>
5. Документация библиотеки socket // URL: <https://docs.python.org/3/howto/sockets.html>
6. Документация библиотеки chess // URL: <https://python-chess.readthedocs.io/en/latest/>

## **ПРИЛОЖЕНИЕ А. ССЫЛКА НА РЕПОЗИТОРИЙ GITHUB**

[https://github.com/Gregory-hub/chess\\_UDP](https://github.com/Gregory-hub/chess_UDP)